

REMARKS

The final office action mailed August 29, 2006 has been carefully considered and these remarks are responsive thereto. Claims 1-5, 7-27 and 29-44 are pending and stand rejected.

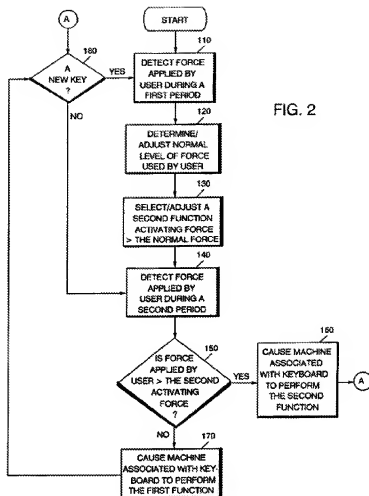
Interview Summary

Applicants wish to thank Examiner Pham and Primary Examiner Chang for their time and courtesy extended to the undersigned during the personal interview conducted October 10, 2006. During the interview, the undersigned and the examiners discussed the present rejection of claims 1 and 17. As to claim 1, the undersigned noted that claim 1 recites a first type message, and that the first type message includes three separate pieces of data: a force update based on updated key force data, a key identifier and a force update indicator. As to claim 17, the undersigned stated that said claim recites receiving registration messages from first and second applications, and that such features are not taught by Barker. Agreement was not reached.

Claim Rejections

The office action rejected claims 1-5 and 7-13 under 35 U.S.C. § 103 based on U.S. Patent 5,675,329 (Barker et al., hereinafter "Barker") or based on Barker in view of what appears to be an English language translation of the abstract of Japanese Patent Publication No. JP05011914A (Tanaka). Applicants respectfully traverse. As discussed at the October 10, 2006 interview, claim 1 recites generation of "first type" messages that include three pieces of data: "generating first type keyboard data messages containing *force updates* based on updated key force data, *key identifiers* for the keys associated with the updated key force data, and *force update indicators*."

The office action relies on the flow chart of Barker figure 2, reproduced below for convenience:



The flow chart of Barker Fig. 2 generally describes an algorithm, taking place inside of a "microcontroller 18" and other keyboard components, for assigning "key scan data bytes" based on the amount of force applied by a user.¹ That microcontroller is situated between a "computer keyboard [11] having an X-Y matrix of momentarily depressed key switches having a conventional QWERTY configuration" and a "system unit of a computer." Barker Fig. 1; col. 2, lines 44 through col. 3, line 20. The microcontroller receives a signal representing pressure exerted on a key (or the entire keyboard); based on that signal, the microcontroller determines

¹ Although blocks 160 and 170 refer to causing a "machine associated with keyboard" to perform a first or second function, Barker teaches that this "causing" is simply the transmission of a key scan data byte to a computer. See *infra*.

what "key scan data byte" it should send to the computer. Microcontroller 18 may be housed within keyboard 18. Col. 3, lines 42-44.

Barker does not teach microcontroller 18 (or a keyboard in which it is housed) sending force data to a computer. The only messages transmitted from microcontroller 18 to the computer are the key scan data bytes.² Even if steps (or transitions between steps) in the algorithm of Barker Fig. 2 could be considered to suggest "messages," Barker does not teach or suggest that any single one of those steps or transitions includes a message having a force update based on updated key force data, a key identifier and a force update indicator. The office action only states the following at page 3:

generating first type keyboard data messages containing force updates based on updated key force data, key identifiers for the keys associated with the updated key force data, and force update indicators (160) in column 4, lines 1-5; and

Barker indicates that in step 160 "a machine associated with the keyboard 11, such as a computer, is caused to perform the second function associated with a key on keyboard 11." Col. 4, lines 3-5. However, another portion of Barker makes clear that step 160 consists of microcontroller 18 sending a "key scan data byte" to a computer. Col. 3, lines 7-13 ("microcontroller 18 produces a control signal prompting the microcontroller 18 to retrieve a key scan data byte corresponding to a second function associated with the key and then sends, in a manner to be described below, the key scan data byte to the system unit 24 of a computer such that the computer is instructed to perform the second function"). There is nothing in Barker suggesting that microcontroller 18 sends key force data or a force update indicator to the computer.

² Barker does refer to microcontroller 18 receiving "a key scan data input signal 26 for prompting the microcontroller 18 to send a key scan data byte to system unit 24." Col. 3, lines 32-34. It is not clear where this signal 26 originates. In any event, Barker does not teach or suggest that signal 26 contains the components of a first type keyboard data message as recited in claim 1. Barker also states that "a user may adjust the threshold value for the second function actuating force stored within microcontroller 18 via the communication link between system unit 24 and microcontroller 18." Col. 3, lines 28-31. Again, there is no teaching or suggestion of a message having a force update based on updated key force data, a key identifier and a force update indicator.

Barker similarly fails to teach or suggest that any other part of the Fig. 2 flowchart includes a single message containing a force update based on updated key force data, a key identifier and a force update indicator. In block 140, for example, a force applied by a user during a second period is detected. Stated differently, another force value is received whenever the algorithm reaches block 140. Nothing in Barker suggests that microcontroller 18 receives some additional piece of information in block 140 signifying that the force data being received is the second (or subsequent) value received for that key. The use made of the force data from block 140 suggests otherwise. Specifically, the force data received in block 140 is compared to "the second actuating force" in block 150. That second actuating force is "greater than the normal force [determined in block 120 from the force received in block 110]... by some predetermined amount" (see col. 3, lines 2-6 and 56-67). That second actuating force is "selected" in block 130 (col. 3, lines 59-61), indicating that the "second actuating force" is stored prior to blocks 140 and 150. Microcontroller 18 would (in block 150) simply compare whatever value is received in block 140 with whatever second actuating force value was selected in block 130. Barker only teaches generation of a new "normal force" for "new" key presses, and new values for the second actuating force would presumably not be stored unless there is a new key press. Microcontroller 18 would need no separate indication in block 140 that the force it is receiving is the second (or subsequent) value for a particular key.³

For at least this reason, claim 1 is allowable. Claim 1 also recites automatically generating, at a repeat rate based on key force data for a key held pressed by a user, a third type keyboard data message indicating the held key has been pressed. The office action concedes that this feature is not taught by Barker. The office action instead relies on Tanaka, and asserts that "it would have been obvious to one with ordinary skill in the art at the time the invention was made to combine the key force data dependent repeat rate taught by Tanaka with the key force sensitive keyboard of Barker in order to 'improve key operability by controlling the automatic repeat speed of keys on a keyboard for which automatic key repeat control is executed.'" Office

³ Although Barker column 4, lines 51-62 describe a way in which the operation of block 120 can be performed based on two forces, it does not teach or suggest a message having a force update based on updated key force data, a key identifier and a force update indicator.

action at page 3. Applicants respectfully note that this argument ignores differences between Barker and Tanaka, and thus fails to provide a sufficient motivation for the proposed combination. Specifically, Barker and Tanaka utilize key force data for different purposes. Barker uses that data to choose between two possible key scan data bytes assigned to a particular key (e.g., lower case letter vs. upper case letter). Tanaka uses the force data to decide how frequently a data value assigned to a key is output. Neither Barker nor Tanaka suggests using key force data for both functions, and the office has not explained how this might be accomplished in the context of Barker and Tanaka. Assume, for example, that the Barker system outputs a lower case letter for a key force less than the "second actuating force" threshold and an upper case letter for a key force above that threshold. If the key force is also used to control repeat rate, does that mean that upper case letters would be repeated and lower case letters would not be repeated (or that upper case letters would be repeated quickly and lower case letters repeated slowly)?

Because the office action has not provided a sufficient motivation to combine Barker and Tanaka, claim 1 is allowable for at least this additional reason.

Claims 2-5 and 7-13 depend from claim 1 are thus allowable for at least the same reasons as claim 1, as well as for additional novel features recited therein. For example, claim 5 recites that receiving keyboard data sets comprises receiving a data set having key identification data and key force data for multiple keys. The office action first asserts that this is taught by Barker figure 2 and by Barker "column 3, lines 50-20." Office action at page 4 (emphasis added). Applicants are not certain which portion of Barker is cited. In any event, nothing in Barker teaches or suggests a data set -- i.e., a *single* data set -- with key identification and key force data for multiple keys. Instead, Barker teaches that key data is received for one key at a time. The office action also cites Barker col. 4, line 22 (office action at 14). However, this portion of Barker is also silent regarding a single data set having key identification and key force data for multiple keys.

The office action also rejects claims 14-16 under 35 U.S.C. § 103 based on Barker or Barker in view of Tanaka. Claim 14 recites receiving a keyboard data set reporting, for multiple

keys of the plurality [of keys] pressed by a keyboard user, key force data and key identification data. The office action does not identify a portion of Barker (or Tanaka) supposedly teaching this feature. In any event, this feature makes claim 14 allowable for at least the same reason as claim 5. Claim 14 further recites associating key identifiers and force values based on the orders in which the key identification data and the key force data appear in the keyboard data set. The office action again cites Barker Fig. 2 and col. 3, "lines 50-20," and argues that "[o]ne can see that the key data are read into the apparatus in a certain order and identified in a certain order." There is nothing in Barker suggesting that key data is processed in any manner other than a key-by-key basis. The "order" would be receive and process key data for a pressed key, receive and process data for the next pressed key, etc.; each data "set" in such a case would apply to one key. There is nothing in Barker suggesting that key data for multiple keys is placed in a single data set, and that such a data set is then parsed to associate key identifiers with force values based on the order in which they appear in that data set.

Claims 15 and 16 depend from claim 14 and are allowable for at least the same reasons as claim 14.

The office action rejected claims 17-21 under 35 U.S.C. § 103 based on Barker. Applicants respectfully traverse. Claim 17 recites, inter alia, receiving a registration from a first application program requesting keyboard input data and key force data, and receiving a registration from a second application program requesting keyboard input data but not requesting key force data. Barker does not teach or suggest these features. Notably, Barker does not contemplate the presence of applications that process key force data. Instead, Barker appears to assume that the operations carried out in connection with Barker figure 2 convert key force data into a format understood by all applications. Barker col. 3, lines 5-28 and 42-44 indicate that the algorithm of Barker figure 2 is performed by a microcontroller 18 within a keyboard, with that microcontroller deciding which key scan code is to be sent (by that same microcontroller) to a "system unit 24 of a computer such that the computer is instructed to perform" a corresponding function.

Because Barker contemplates determining how to interpret key force data before that data even reaches a computer, there is no need for an application in that computer to request key force data. Stated differently, Barker is directed to hardware that determines key force and then selects a key code based on that force; the hardware then sends that key code to the computer. There would be no reason for an application in the computer to request force data, as the hardware has already determines what the key force data means.

The office action concedes at page 10 that Barker fails to teach the recited steps of receiving a registration from a first application and receiving a registration from a second application. The office action then argues the following:

However, in Fig. 2 and in column 3, lines 50-20; Barker teaches of the process that has various prompts to take in keyboard data because this has a similar function to the applications as in the claim limitations, the prompts can be seen and viewed as separate applications.

Applicants respectfully submit that this argument is not correct. Barker says nothing about "prompts" in the flowchart of Fig. 2. The blocks in that flowchart simply reflect steps of a single algorithm. Steps of that algorithm are not "applications," much less two separate applications.⁴ Even if the individual steps of the algorithm in Barker Fig. 2 could be considered applications, however, Barker in no way suggests that those steps register with other parts of the algorithm so that they can receive data.

The office action further states that:

It would have been obvious to one with ordinary skill in the art at the time the invention was made to include application software with the apparatus in order to associate a different task or data with a different application.

⁴ "Application program" has an accepted meaning in the art as a computer program that performs some function for an end user (or perhaps for another application program). See, for example, the searchwebsiteservices.com definition of "application" <http://searchwebsiteservices.techtarget.com/sDefinition/0,290660,sid26_gci211585,00.html>. Application programs use services of system programs, operating system programs, etc. Nonlimiting examples of application programs included in paragraph [31] of Applicants' specification include word processing programs, games, and e-mail or internet chat clients.

Applicants also respectfully suggest that is incorrect, and unsupported by Barker or any other authority cited by the office action. As discussed above, the entire teaching of Barker is premised upon a hardware implementation in which a microprocessor inside a keyboard determines how force on a key should be translated into a key scan data byte, with that key scan data byte then sent to a computer. The office action gives no indication why a person of skill in the art would have been motivated to completely redesign Barker so that the assignment of a key code based on a force occurs in an application inside a computer.

Finally, the office action states:

Since the apparatus of Barker is able to detect and communicate data information in regards to both (1) which key is pressed (see column 3, line 55) and (2) the key force data (see column 4, line 5); then it would be known in the art to provide an apparatus that is able to only send only one set of data (either the which key is pressed or the key force data).

It would have been obvious to one with ordinary skill in the art at the time the invention was made to combine only send one set of data, such as the keyboard input instead of both the keyboard input and the key force data in order to only provide needed data needed by various application to avoid unnecessary steps.

The above argument relies on an incorrect assumption. Specifically, Barker does *not* teach that the described apparatus is able to communicate information in regards to both which key is pressed and key force data. As explained above, microcontroller 18 assigns a key scan data byte based on the applied force, but only that key scan data byte (and *not* the force itself) is sent to a computer.

Because Barker does not teach all features of claim 17, and because the office action has failed to provide a motivation for modifying Barker so as to teach all of those features, claim 17 is allowable. Claims 18-21 depend from claim 17 and are allowable for at least the same reason as claim 17, and further in view of additional features recited therein.

The office action rejected claim 22 under 35 U.S.C. § 103 based on Barker in view of U.S. Patent 5,576,734 (Danielle et al., hereinafter "Danielle"). Claim 22 depends from claim 17,

and Danielle fails to teach the features of claim 17 not taught by Barker. Accordingly, claim 22 is also allowable.

Claims 23-27 and 29-44 recite a computer-readable medium having instructions for performing methods similar to those recited by claims 1-5 and 7-22, respectively, and are allowable for the same reasons as claims 1-5 and 7-22.

Respectfully Submitted,

By: **/H. Wayne Porter/**

H. Wayne Porter
Registration No. 42,084

BANNER & WITCOFF, LTD.
1001 G Street, N.W., 11th Floor
Washington, D.C. 20001
Office: (202) 824-3000

Dated: October 30, 2006